



Fig. 1

Dynamic Data Defining

Supports the dynamic definition of data files for:

- **Data syntaxes:**
 - Logical flat Files
 - Delimited flat files
 - Stream data format files
 - XML data
- **File Structure and Content:**
 - Hierarchical record/segment structure and content
 - Segment Grouping
 - Minimum/maximum record/segment occurrences
- **Segment/record data content:**
 - Alphanumeric data element characteristics
 - Numeric data elements with implied decimal point notation
 - Numeric data elements with explicit decimal point notation
 - Numeric data elements with embedded sign
 - Date element formats
 - Time Element formats
 - Binary elements
 - Encoded elements and automatic code conversion

Fig. 2

Standard Conversions

Data is converted on a one to one basis, enabling supporting the following data re-formatting capabilities:

- Stream data to stream data with data optimisation (example: exclude surplus leading zeros and trailing spaces)
- Stream data to logical flat file
- Stream data to delimited flat file
- Stream data to XML

- Logical flat file to stream data with data optimisation
- Logical flat file to logical flat file
- Logical flat file to delimited flat file
- Logical flat file to XML

- Delimited flat file to stream data with data optimisation
- Delimited flat file to logical flat file
- Delimited flat file to delimited flat file
- Delimited flat file to XML

Fig. 3

Customised Conversions

Supports the development of conversions to meet customer bespoke requirements by providing run-time translation rules.

The currently supported rules provide the following functions:

- Invoke a special bespoke alternative translation processing routine
- Invoke special input data processing routines
- Invoke special output processing routines
- Accumulate an input numeric value.
- Construct output segment using data from multiple input segments
- Arithmetic Computations
- Conditionally or unconditionally ignore an input segment.
- Conditionally or unconditionally ignore a complete input message
- Conditionally create an output segment
- Establish and maintain counts
- Test for specific data conditions and execute or bypass rules depending on a true/false result
- Invoke rejection of a complete input data file (usually when a specific data condition is detected)
- Store data element contents for later use
- Turn switches on or off to record specific conditions
- Edit input data to construct output data
- Use system date and time to provide data
- Set up binary data
- Use stored data

Fig. 4

Sample Rules Implementation #1	
Rules for Automobile Loss Notice processing from ACORD standard forms	EasyLink Action
RqUID	<ol style="list-style-type: none"> 1. If the field is present, it will consist of Lowercase Hex Digits (0-9,a-f) and dashes. The expected format is: "1b1bbhhh-hhhh-hhhh-hhhh-hhhhhhhhhhh" where "h" is a hex digit. 2. If no Id is present, then Easy link should generate one. 3. An Easylink generated Id, should consist "1b1be000" as the first set of hex digits, followed by two sets of 4 Easylink defined digits, (all zeros are fine for Easylink digits if Easylink has not used this for custom tracking information) followed by a 16 digit GUID (Globally Unique Identifier) that uniquely identifies the message (hhhh-hhhhhhhhhhhh).
"street address" lines, Addr1 and Addr2	<ol style="list-style-type: none"> 1. If there are multiple "Address Lines" (i.e. lines between the name and the city/state/zip code), they should be concatenated into a comma separated string and stored in the Addr1 element. 2. If multiple names are provided – i.e. John and Jane doe – "John and Jane" should be considered the first name. 3. Street portion of the address is limited to 55 characters. 4. City is limited to 25 characters. 5. [20010] Company Name or Last Name is required. If neither is present, then the GeneralPartyInfo should not be captured

Fig. 5A

Sample Rules Implementation #1	
Rules for Automobile Loss Notice processing from ACORD standard forms	EasyLink Action
Date fields should be assumed to contain a month, day, and year value.	<ol style="list-style-type: none"> 1. If numeric month and day are specified, the month can be assumed to occur first. 2. If a two digit year is supplied, assume values between the current two digit year and 99 – currently 03 - (inclusive) to be in the 20th Century and values between 0 and the previous two digit year – currently 02 - (inclusive) to be in the 21st Century, except where noted as otherwise. 3. If month text is used, the month can be assumed to be in US English. 4. Dates will be assumed to be Eastern Standard Time.
Global rules: With the exception of the identified required fields, any field that fails validation:	<ol style="list-style-type: none"> 1. Note in the RemarkText Claim element and passed as a blank value.
Date of Loss	<ol style="list-style-type: none"> 1. [20020] Must be equal to or less than the date the fax is received.
Form.ABCDEd Policy #	<ol style="list-style-type: none"> 1. If the Policy Number is captured, use the following Underwriting Company table to determine the ABCDEd. [table deleted] Position1 of the Policy Number is the Company Code. 2. If the Policy Number is not captured, capture the ABCDEd value from the Form.ABCDEd field. 3. If the ABCDEd is still not captured is not captured, match the name in the Form.Company field to the Underwriting Company table to lookup the ABCDEd.
Form.Curent.Date	<ol style="list-style-type: none"> 1. Date must be equal to or less than date fax was received. 2. Date must be equal to or greater than Form.Loss.Date.
Insured.SSN.Or.FEIN	Must be a social security number. 9 digits, optionally separated as follows by spaces or dashes: 000-00-0000. It should be passed in the ACORD Message as 9 numeric digits, no spaces or dashes
ClaimsParties Reconciliation	<ol style="list-style-type: none"> 2. If any ClaimsParty on the form contains the case insensitive word "Insured" as the name, then add the appropriate role to the insured. 3. If Insured Vehicle on the Auto form is blank, than the insured is the owner. 4. If any driver field contains the case insensitive word "same", the vehicle owner is the driver. 5. If a ClaimsParty field contains no address and has the same First, Last, and Middle name as another ClaimsParty, they are the same person. 6. If a ClaimsParty field contains an address and has the same First, Last, and Middle name and the same address as another ClaimsParty, they are the same person.
Contact	<ol style="list-style-type: none"> 1. If contact contains "same" or "insured" (case insensitive) the Insured is also the contact. 2. Contact Name should not be captured unless there is a Contact Phone Number and/or Contact Address provided.
Claimant Identification	<ol style="list-style-type: none"> 1. The remainder of these rules should be used to identify the claimants. If no claimant can be identified, then "missing required field" exception should be thrown. 2. If (Pol.Coll.Ded and/or Pol.Other.Than.Coll.Ded) is populated with a valid amount and InsVeh.Describe.Damage is not empty and contains anything except a case insensitive "No Damage" value - "none", "n/a", "no damage", "none reported", or zero as integer, float, or currency - set the insured as the claimant. 3. If PropDam.Owner.CompanyName or PropDam.Owner.Name.Last is

Fig. 5B

Sample Rules Implementation #1	
Rules for Automobile Loss Notice processing from ACORD standard forms	EasyLink Action
	<p>populated, make that ClaimsParty the claimant. "Unknown" is a valid PropDam.Owner.Name.Last.</p> <p>4. If one of the injured fields contains the Insured or one of the injured fields contains only the first name of the insured or the case insensitive word "insured" make the insured the claimant.</p> <p>5. If the injured field contains a participant and INSVEH or PED is checked make that ClaimsParty the claimant.</p> <p>6. [21010] If claimant cannot be identified.</p>
Form.Policy.Number	[21020] Must be 10 characters with the first 3 being "XYZ" and the last six being numeric. The fourth character can be alpha or numeric (A-Z, 0-9): XYZcnnnnnn
Form.Policy.Expiration.Date	<p>1. For two digit years, years between 0 and 8 years less than the date of the loss should be considered in the 21st Century, remainder should be considered in the 20th.</p> <p>2. [21040] Must be no more than 3 years after the Form.Policy.Effective.Date</p>
Form.Time.Of.Loss	[21050] In addition to being a valid time, one of Form.Time.Of.Loss.AM or Form.Time.Of.Loss.AM must be checked or AM or PM specified as part of the time. This is used to convert time to ACORD 24h time. If both or none are checked, the time should not be captured.
InsVeh.Veh.Number	[21060] Vehicle Number is an integer identifying the vehicle on a policy. It will be specified using 2 numeric digits and will fall between 01 and 99.
InsVeh.Driver.DOB	[21080] Must be less than the date of loss.
Loss.Loc.Of.Accident	Captured even if address cannot be parsed into Addr specifics.
InsuredVeh.Owners.CompanyName and InsuredVeh.Owners.Name	Only one of these should be present.
Injured.Ped and Injured.InsVeh and Injured.OthVeh	If none of these is selected, default to Injured.InsVeh.
Message Transfer Specification	POST request over HTTPS
Message Format	ACORD XML w/TIF
Response Codes	Success, Redirect, Unauthorized access, other errors w/exception handling

Fig. 5C

Sample Rules Impelementation #2	
Rules for Custom order processing	EasyLink Action
Branch # = 3 numerics	If present, validate against branch list, then continue processing
Order # = 6 numerics	If present, log as an "order"; create index file and a directory for accompanying image files
<p>The label of the index file to denote the content of the order</p> <p>Sample: Three faxes are received for the same order (123000001)</p> <p>Fax#1:</p> <p>Page 1: Cover Page (type DR)</p> <p>Page 2: Non-Form</p> <p>Page 3: Non-Form</p> <p>Fax#1 :</p> <p>Page 1: Cover Page (type BP)</p> <p>Page 2: Non-Form</p> <p>Page 3: Non-Form</p>	<p>Create Index: A123000001.NDX</p> <p>BrnOrder :123000001</p> <p>Page Cnt :2</p> <p>File Name001:A123000001001_DR.tif</p> <p>File Name002:A123000001002_DR.tif</p> <hr/> <p>Page Cnt :3</p> <p>File Name003:A123000001001_BP.tif</p> <p>File Name004:A123000001002_BP.tif</p> <p>File Name005:A123000001003_BP.tif</p>

Fig. 6A

Sample Rules Implementation #2	
Rules for Custom Order Processing	EasyLink Action
Page 4: Non-Form	
<p>Successful orders to have image directory in following format: Directory Name: <ZZZOrd#> Cover Page File Name: A<ZZZOrd#>000_<type>.CV<SeqLetter> Attachment File Name: A<ZZZOrd#><PgNum>_<type>.TIF</p>	<ol style="list-style-type: none"> 1. Assign Order # as directory name 2. Assign Order # plus '000', file type, .CV<SeqLetter> as file name for cover page 3. Assign order # plus page number starting with '001' plus file type.TIF as file name for each attachment <p>Create Directory with attachments: A123000001 (the following files are listed in the alphabetic order) A123000001000_BP.CVA (Fax#2, Page#1) A123000001000_DR.CVA (Fax#1, Page#1) A123000001001_BP.TIF (Fax#2, Page #2) A123000001001_DR.TIF (Fax#1, Page #2) A123000001002_BP.TIF (Fax#2, Page #3) A123000001003_BP.TIF (Fax#2, Page #4)</p>
<p>Unsuccessful orders, one cover page only fax [ONE], create an exception directory in the following format: Tag file: <TID>[_SeqID].<ExceptionCode> Directory: < TID>[_SeqID] Files under Directory (for OCR & NOF): Page_<PageNumber>.TIF Files under Directory (for ONE): A<ZZZOrd#>000.TIF Where: <TID> = EasyLink Transaction ID, in the format of 'Mnn...n' [_SeqID] = Needed only if there are multiple exceptions for the same TID<ExceptionCode></p>	<ol style="list-style-type: none"> 1. operator OCR's cover page data and adds ONE to data header 2. EasyLink assigns file names to data, e.g. Fax: Page 1: Cover Page (with ONE exception) Exception: Tag File: M22000000000000000002.ONE Dir: M22000000000000000002 Files in Dir: A123000001.TIF
Routing instructions	<ol style="list-style-type: none"> 1. XML format with TIFF image attachments 2. XML Tags: <TID>, Incoming fax number, time/date stamp, Directory name, # files, #pages

Fig. 6B

Schedule 0001

Events

```
NewFax (/render, priority = 1, success = NewImage, fail = QueueFax)
QueueFax (/render, priority = 0, success = NewImage, fail=/email (renderfail);/alarm (renderfail))
NewImage (/ocr, priority = 1, success = NewCsv, fail = QueueImage)
QueueImage (/ocr, priority = 0, success = NewCsv, fail = /email (ocrfail); /alarm (ocrfail))
NewCsv (/trans, priority = 1, success = NewFile, fail = QueueCsv)
QueueCsv (/trans, priority = 0, success = NewFile, fail = /email (transfail); /alarm (transfail))
NewFile (/arc, priority = 1, success = NewOutdoc, fail = QueueFile)
QueueFile (/arc, priority = 0, success = NewOutdoc, fail = /email (arcfail); /alarm (arcfail))
NewOutdoc (/deliver, priority = 1, success = /email (deliver_notice), fail = QueueOutdoc)
QueueOutdoc (/deliver, priority = 0, success = /email (deliver_notice), fail = /email
(non_del_notice); /alarm(deliverfail))
```

Program Parameters

```
Render Options = tiff, fine
Email Address = customer_name@mail.com
Trans Source File Type = csv
Trans Target File Type = XML
Trans Business Rules File = customerx_rules_file
Arc Period = 60 days
Delivery Protocol = FTP
Content Routing (
  If Policy # = 3100xxxxxx, IP Addr = nnn.nnn.nnn.nnn
  Elseif Policy # = 3200xxxxxx, IP Addr = nnn.nnn.nnn.mmm
  Else IP Addr = nnn.nnn.nnn.ooo)
```

Fig. 7

Results that match your search criteria:

Origin	Destination	Received	Status	Delivered	Pages	Transaction ID	Attempt	Service
<u>10280877</u>	<u>612601048</u>	9/15/2003 22:58:08	successful	9/16/2003 22:59:01	3	<u>2591331500</u>	1	Document Acceptance
<u>10280877</u>	<u>612601048</u>	9/15/2003 22:59:01	successful	9/16/2003 23:18:04	3	<u>2591331500</u>	1	Document Capture
<u>10280877</u>	<u>612601048</u>	9/15/2003 23:18:05	successful	9/16/2003 23:21:03	3	<u>2591331500</u>	1	Data Translation
<u>10280877</u>	<u>612601048</u>	9/15/2003 23:21:03	successful	9/16/2003 23:23:04	3	<u>2591331500</u>	1	Data Archive
<u>10280877</u>	<u>612601048</u>	9/15/2003 23:23:04	failed	9/16/2003 23:25:06	3	<u>2591331500</u>	1	Document Delivery

Fig. 8